

# Be Suspicious about Trusting Unauthenticated External Representation of Internal Data Structures

---

William L. Fithen, Software Engineering Institute [vita<sup>3</sup>]

Copyright © 2005 Carnegie Mellon University

2005-10-03

L4 / D/P<sup>4</sup>

Trusting unauthenticated externalized data structures can introduce vulnerability.

## Description

Many systems depend on technologies that support the rendering of internal data structures or objects into the form of *serial streams* of bytes. These external representations are used for a variety of functions, including mobile objects, distributed objects, and remote procedure calls. When an adversary can modify the serial stream between its time of production and later use, a vulnerability may exist. Four different situations can cause vulnerability:

- If the altered serial stream is syntactically invalid, blind restoration of the serial stream can result in a variety of vulnerabilities, the most frequent being buffer overflow.
- If the altered serial stream is syntactically invalid and the restoring program recognizes this, it will not restore the stream, avoiding more serious vulnerabilities but perhaps still resulting in a denial of service.
- If the altered serial stream is syntactically valid but semantically invalid, restoration of the serial stream results in a valid internal representation of data structures or objects, but those structures or objects are not what the original serializing program intended to have restored. This can result in a variety of erroneous or vulnerable behaviors.
- If the altered serial stream is syntactically valid but semantically invalid and the restoring program recognizes this, it will not restore the stream, avoiding more serious vulnerabilities but perhaps still resulting in a denial of service.

## References

- [VU#192995]      Havrilla, Jeffrey. *Vulnerability Note VU#192995: Integer overflow in xdr\_array() function when deserializing the XDR stream.* <http://www.kb.cert.org/vuls/id/192995> (2005).
- [VU#597889]      Dougherty, Chad. *Vulnerability Note VU#597889: Microsoft COM Structured Storage Vulnerability.* <http://www.kb.cert.org/vuls/id/597889> (2005).

## Carnegie Mellon Copyright

---

Copyright © Carnegie Mellon University 2005-2010.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu)<sup>1</sup>.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

NO WARRANTY

---

3. [http://buildsecurityin.us-cert.gov/bsi/about\\_us/authors/320-BSI.html](http://buildsecurityin.us-cert.gov/bsi/about_us/authors/320-BSI.html) (Fithen, William L.)

1. <mailto:permission@sei.cmu.edu>

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.